



Classes - Wrapping it up

Writing a class from scratch



Time to wrap it up

- You are now done with the chapters on objects and classes
- It is time for an exercise
- Your task is to write a class from scratch

Time for the Time class

You should write a class `Time`, for creating `Time` objects representing a point in time (on a 24 hour clock).

- The class should represent a time with hours, minutes and seconds
- It should be possible to create a new `Time` from three `int` values representing hour, minute and second
- The class should have an instance method `toString()` which returns a `String` representing the `Time`
- The `String` should have the format `HH:mm:ss` - for instance `23:49:59` (24 hour time)

continued...

- The class should have an instance method `tick()` which increments the Time with one second
 - That method is here, of course, mostly for you to practice how to increment the Time correctly with wrapping around values:
 - 59 seconds + 1 increments the minute by one and sets the second to 0
 - 59 minutes plus one minute increments the hour with one and sets the minute to 0
 - 23 hours plus one wraps around to 00 (midnight)
- The class should have an instance method for setting the Time to a completely new Time (accepting three ints as arguments for hour, minute and second)

continued...

- The class should have accessor methods returning the int values for:
 - hour
 - minute
 - second
- You should use, internally, one single int holding the seconds representing the time:
 - a number between 0 (00:00:00) and 86399 (23:59:59) as it happens
- That is the number of hours times 60 times 60 plus the number of minutes times 60 plus the number of seconds

Object oriented aspects on the class

- This class will serve as an example of the thing called encapsulation - an object has both state and behavior
- It also shows the concept of information hiding in that it has a private internal representation of the time and opens up abstractions for manipulating and accessing the values of the time represented
- How we store the actual data for the Time the object represents is of no concern of the client code using the Time
- Any code using our Time class will have to do with the public interface to the object of type "reference to Time"
 - The public interface is just a fancy word for the public constructor and the public methods.

This is an exercise

Again, this class is meant as an exercise for you, and not necessarily as an example of a well-designed typical Java class - we have chosen it because we think it contains interesting elements of writing classes (private variables and public methods etc).

Some maths in the exercise

We wanted to show you that the internal representation of an object may differ from the public interface of the object.

Even though the constructor and the setTime method accepts three int values (representing hour, minute and second), the object may very well use something else to store the time.

In this example, we use one int to store the total number of seconds any time between 00:00:00 and 23:59:59 has. In order to extract only the hours from this internal (private) int, we need to calculate “backwards”.

Some maths in the exercise

So, your int will store a value between 0 and 86399, where 0 represents the time 00:00:00 (since that time “has no seconds” as it is the start time), and where 86399 represents 23:59:59 since that time has:

- 23 x 3600 seconds in the hours part
- 59 x 60 seconds in the minutes part
- 59 seconds in the seconds part

$$23 \times 3600 + 59 \times 60 + 59 = 86399$$

Now, given any value for the total seconds, how do we calculate hours, minutes and seconds?

Some maths in the exercise

Let's say we have stored the time as 3730 seconds, and now we want to calculate the seconds the time represents. We could do like this:

There are 60 whole minutes in 3730, using integer division:

$$3730 / 60 = 62.$$

$$62 * 60 = 3720 \text{ seconds.}$$

$$3730 - 3720 \text{ leaves us } 10 \text{ seconds.}$$

Answer: the seconds part of the time represented by 3730 is 10.

Some maths in the exercise

$$3730 / 60 = 62.$$

$$62 * 60 = 3720 \text{ seconds.}$$

$$3730 - 3720 \text{ leaves us } 10 \text{ seconds.}$$

The above can be calculated using the % operator (remainder, or modulo):

$$3730 \% 60 = 10.$$

The remainder calculates “what is left after integer division”.

A simpler example of remainder

Let's say you have 17 apples and want to distribute them evenly among 4 horses. How many apples will you have left?

We know that 17 divided by 4 is 4, using integer division. 4 times 4 is 16, so we'll end up with one apple remaining.

As it turns out, using the remainder operator, you will always get a number between 0 and the right-hand operand minus one. So anything % 4 is always a number between 0 and 3.

If you get 0 (no apples remain), the left-hand operand is divisible by the right-hand operand.

We chose 60 as the right-hand operand

Any number % 60 is a number between 0 and 59, which is exactly what we want. Another way to put it, is to repeatedly take 60 from the number until we have a number less than 60.

Our total amount of seconds variable works this way, since we calculate it using multiples of 60 plus the seconds part. The hours passed to the constructor is multiplied by 3600 ($60 * 60$), the minutes are multiplied with 60 and the seconds are added at the end.

Calculating the minutes

When we want to calculate the minutes our total number of seconds represent, what do we need to do then?

First we find out how many whole minutes, disregarding the seconds, our total number of seconds contain. Again, let's take 3730 as the example time.

3730 seconds are 62 whole minutes ($3730 / 60$, using integer division). Now, we want to limit ourselves to a number between 0 and 59, because we can't have e.g. a clock showing 62 in the minute field! That should mean one our and two minutes, right? So we take $62 \% 60$ and we get 2.

$3730 / 60 \% 60 = 2$. Answer: the minute field is 2.

Calculating the hours

Again, let's take 3730 as the example total amount of seconds in a time. How many hours is that? Well, one hour is 3600 seconds, so here integer division with 3600 works well:

$$3730 / 3600 = 1.$$

Answer: The hour part of 3730 is 1.

Summary

So, internally a Time object stored a time as e.g. 3730 total seconds. We could calculate the values for the hour(), minute() and second() accessor methods:

- Second: $3730 \% 60 = 10$
- Minute: $3730 / 60 \% 60 = 2$
- Hour: $3730 / 3600 = 1$

Does it add up? Counting backwards: the time 01:02:10 should be stored as:

$$1 * 3600 + 2 * 60 + 10 = 3730.$$

It works :-)

Another use for remainder

Let's say we have a large integer, and want to extract the last 1, 2, and three digits. How do we do that? We could convert the integer to a String and use String manipulation, but that's cheating!

Let's take this large number: 123456789

Last digit is $123456789 \% 10$

Two last digits are $123456789 \% 100$

Three last digits are $123456789 \% 1000$ (etc)

Can you explain why this works?

Getting started

You'll find detailed instructions on our wiki here:

[http://wiki.juneday.se/mediawiki/index.php/Java Classes Wrapping it up](http://wiki.juneday.se/mediawiki/index.php/Java_Classes_Wrapping_it_up)